

Aberystwyth University

Replicating complex agent based models, a formidable task

Donkin, Elizabeth; Dennis, Peter; Ustalakov, Andrey; Warren, John; Clare, Amanda

Published in:

Environmental Modelling and Software

DOI:

[10.1016/j.envsoft.2017.01.020](https://doi.org/10.1016/j.envsoft.2017.01.020)

Publication date:

2017

Citation for published version (APA):

Donkin, E., Dennis, P., Ustalakov, A., Warren, J., & Clare, A. (2017). Replicating complex agent based models, a formidable task. *Environmental Modelling and Software*, 92, 142-151.
<https://doi.org/10.1016/j.envsoft.2017.01.020>

Document License CC BY

General rights

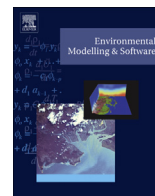
Copyright and moral rights for the publications made accessible in the Aberystwyth Research Portal (the Institutional Repository) are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Aberystwyth Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Aberystwyth Research Portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

tel: +44 1970 62 2400
email: is@aber.ac.uk



Replicating complex agent based models, a formidable task



Elizabeth Donkin^{a,*}, Peter Dennis^a, Andrey Ustalakov^{b,1}, John Warren^{a,2},
Amanda Clare^b

^a Institute of Biological Environmental and Rural Sciences (IBERS), Aberystwyth University, Pengllys Campus, SY23 3DA, UK

^b Department of Computer Science, Aberystwyth University, Pengllys Campus, SY23 3DB, UK

ARTICLE INFO

Article history:

Received 1 December 2016

Received in revised form

20 January 2017

Accepted 27 January 2017

Keywords:

Agent-based model

Individual-based model

Model replication

NetLogo

Open code

Repast

ABSTRACT

Promoting replication of models is unarguably a positive step for agent based modelling, as replication promotes rigorous testing. Model replication remains rare, yet is vital to assessing the repeatability of existing agent based models. Notably, more work is needed to assess cross platform and language replication, which represent potential sources of variability between model results. An existing, complex agent based model was replicated using two widely used platforms (NetLogo and Repast). When results generated by the models were compared, the findings differed not only in magnitude but the trends produced by the data, resulting in different conclusions being drawn from each set of model predictions. The variation between the models is believed to be a result of the complexity of encoding a substantial theoretical model in particular programming languages. This highlights the express need to document replication of existing models in order to fully understand the potential limitations to replication.

© 2017 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Software availability

Name of the Model: Repast Model (plantInsectPolyculture)

Developers: Andrey Ustalakov, CA Technologies, Oxford, UK;

Elizabeth Donkin, IBERS, Aberystwyth University, UK.

Software Required: Repast Symphony 2.2.0

Program Language: JAVA

Code Availability: <https://github.com/lizzydonkin/plantInsectPolyculture>

Platform Availability: free download <http://repast.sourceforge.net/download.html>

Name of the Model: NetLogo Model (plantInsectPolycultureNetLogo)

Developer: Elizabeth Donkin, IBERS, Aberystwyth University, UK.

Software Required: NetLogo 5.1.0

Program Language: NetLogo

Code Availability: <https://github.com/lizzydonkin/plantInsectPolycultureNetLogo>

Platform Availability: free download <https://ccl.northwestern.edu/netlogo/download.shtml>

1. Introduction

Agent based models (ABM) are widespread throughout the biological, ecological and environmental sciences (Bithell and Brasington, 2009; Bradhurst et al., 2016; Grimm, 1999; Grimm et al., 2005; Matthews et al., 2007). They provide valuable insight into often complex agent based systems and have become an indispensable research tool (Thiele and Grimm, 2015). Despite the value of ABMs, developed models are rarely repeatedly applied. The majority of researchers develop models from scratch rather than testing and building on the work of others (Thiele and Grimm, 2015; Wilensky and Rand, 2007).

The importance of developing a culture of model replication in the sciences is an idea gaining traction (Axtell et al., 1996; Collins et al., 2015; Thiele and Grimm, 2015; Wilensky and Rand, 2007). Replication exposes models to external scrutiny and facilitates robustness analysis that is rarely implemented by the original model developers. This results in more reliable models that have a body of literature supporting their predictions (Axelrod, 1997; Axtell et al., 1996; Easterbrook, 2014; Thiele and Grimm, 2015).

Replication and reuse of models is unarguably a positive

* Corresponding author.

E-mail addresses: eld38@aber.ac.uk, donkin.e@gmail.com (E. Donkin).

¹ Present address: 11 Oasis Business Park, Eynsham, OX29 4TP, UK.

² Present address: University of Natural Resources and the Environment, Vudal Campus, East New Britain, Papua New Guinea.

development for agent based modelling (Easterbrook, 2014; Thiele and Grimm, 2015). However, in the current research environment there are complications hampering a wide spread adoption of model replication. Agent based models representing natural systems can be, and often are, highly complex (Evans et al., 2013; Sun et al., 2016; Thiele and Grimm, 2015). To facilitate replication the documentation associated with complex models needs to be detailed, and source code made freely available. This level of information is often lacking for published ABMs and although detailed frameworks for model descriptions exist and repositories for code have been established, models are still published without this information (Rollins et al., 2014; Thiele and Grimm, 2015).

As the importance of replication is realised, there is a real need to address whether it is practically possible to replicate a complex model from the information typically provided. There is also a real need to address how and when replication can and should be implemented, and highlight potential pitfalls (Axtell et al., 1996; Hales et al., 2003). One such complication can be cross platform and language replication. In an ideal world, every researcher in agent based modelling would adopt the same programming language and development platform. However, this is unrealistic. Researchers will adopt a method of programming and developing ABMs that suits their ability and resources. Programming languages and technology evolve relatively quickly and current languages may not be the most popular, effective or practical in the near future (Meyerovich and Rabkin, 2013).

There are few publications documenting ABM replication across platforms and programming languages, or indeed documenting model replication at all. Those that have been published have dealt with relatively simple models (Axtell et al., 1996; Bajracharya and Duboz, 2013; Railsback et al., 2006; Wilensky and Rand, 2007). Replication of these simple models across platforms has been found to be problematic in practical terms, making the process of replication a lengthy and complex task (Wilensky and Rand, 2007). However, of greater consequence is the indication that the process can introduce variation in results through disruption of the original model design when made to fit a different language (Bajracharya and Duboz, 2013). This could potentially prevent the successful replication of complex models, as investigating all possible interacting consequences of coding choices would be a near impossible task.

Here we attempt replication and testing of a previously published agro-ecological, agent based model. The model simulates pest insects within an agricultural setting. Insect behaviour is regularly modelled using ABM, due to the ability to incorporate fine scale movement behaviour (Almeida et al., 2010; Perez and Dragicevic, 2010). This particular model was chosen as relevant to our research, and we wished to adapt it to further explore the parameter space. However, the model is typical of many complex ABMs simulating multi-species interactions. We replicated the model based on the information provided in the published article, using two established agent based modelling platforms. The process of replication is discussed, with particular focus on the potential pitfalls. Source code for the two reproduced models are presented for further use and development.

2. Materials and methods

An agent based model, designed and implemented by Potting et al. (2005) was recreated in two agent based modelling platforms: Repast Symphony 2.2.0 and NetLogo 5.1.0. The original model was developed in Visual Basic and the original source code is not available (R Potting, personal communication). To assess how accurately the model could be replicated from the existing information, the original model was recreated as closely as possible

using NetLogo, with only the information provided in Potting et al. (2005). To assess the potential for translational differences between models the NetLogo version of the model was recreated in Java within Repast Symphony, using the NetLogo code and description. The model replications were implemented by the same authors.

Repast Symphony is a group of free and open source modelling platforms, that make use of inter-language libraries (North et al., 2013). The Repast environment provides a graphic user interface whilst users develop models in a general purpose programming language such as Java, as was used in our replication. NetLogo is also a free and open source simulation environment that utilises a programming language, also named NetLogo, designed specifically for developing ABMs (Tisue and Wilensky, 2004). Models are developed in a graphical environment where the user controls individual agents, termed 'turtles' in the NetLogo literature, in a grid environment of patches (for detailed reviews of the platforms see: Tisue and Wilensky, 2004; North et al., 2013).

The original model description is available in Potting et al. (2005). Our interpreted, replicated NetLogo and Repast version is described below using the ODD protocol outlined by Grimm et al. (Grimm et al., 2010, 2006). Assumptions and discrepancies with the original description are outlined.

2.1. Model description

2.1.1. Overview

2.1.1.1. Purpose. The purpose of this model is to understand how insect behavioural ecology, in regards to foraging ability and behaviour of different pest insect species, could impact the efficacy of polyculture planting strategies for preventing damage to crops from pest insect species.

2.1.1.2. State variables and scales. The model simulates time as discrete time steps with a fixed duration. The model environment is comprised of a grid of 100×100 cells with a fixed location, each cell representing a plant of a certain type. Each grid cell, or plant, can be occupied by an infinite number of insects, or remain unoccupied. Upon model initialisation the environment is created with a given proportion of each plant type and a given spatial arrangement.

The plant types are distinguished by the variable 'plant specific flight tendency' and whether their quality can be detected by an insect in flight. 'Plant specific flight tendency' describes the probability of an insect leaving the plant to forage or remaining and feeding. Low values of 'plant specific flight tendency' describe a favourable plant and vice versa. Plant types are defined in Table 1. State variables for the global environment, insects and plants are summarised in Table 2. The three possible spatial arrangements of the crop and non-crop plants are described in Fig. 1.

Insects within the model can have one of three search strategies: visual, olfactory or contact. The starting density of the population is 500 insects, all of which are univoltine and are identical in their search strategy.

2.1.1.3. Process overview and scheduling. The following processes

Table 1

Plant types and their corresponding values for 'plant specific flight tendency'.

Plant type	Plant specific flight tendency	Detected in flight
Favourable		
crop	0.5	yes
trap crop	0.05	yes
Unfavourable		
deterrent	1	no
repellent	1	yes

Table 2

An overview of state variables and the associated default and alternative values, for the global environment, plants and insects. Adapted from Potting et al. (2005).

Parameter	Default	Alternative
Global environment		
Grid size	100 × 100	N/A
Proportion non-crop vegetation	0.25	N/A
Spatial configuration	Intercrop	Border, Randomly distributed patches
Insect population size	500	N/A
Initial distribution of insects	Airborne (random)	East, West
Time steps	50	N/A
Runs per scenario	40	N/A
Insect mortality	0.05	N/A
Insects		
Maximum movement length	10	1–20
Emigration tendency (<i>e</i>)	0.05	N/A
Sensory mode	Olfactory	Contact, Visual
In flight perception probability (olfactory searchers only)	0.7	N/A
Memory size	5	N/A
Plants		
Plant specific flight tendency (<i>f</i>)	0.5	0.05, 1
Damage threshold	100	1–100
Cumulative herbivore days (CHD)	0	N/A

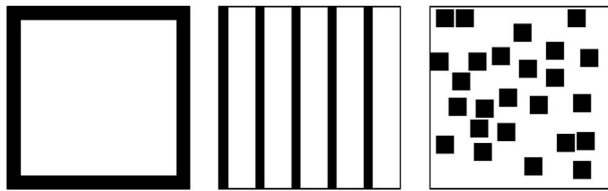


Fig. 1. Possible spatial arrangements of crop (white) and non-crop plants (black) in the 100 × 100 grid environment. Each arrangement contains 75% crop and 25% non-crop plants. Taken and adapted from Potting et al. (2005).

are defined in the model:

Natural mortality. An age dependent mortality rate that is applied to all insects at the beginning of each time step.

Initiation of foraging. The initial decision to stay on the current plant, conduct a local search or make a longer foraging flight.

Foraging flight. The process of the insects moving from their current plant and seeking the next favourable plant to move to, this is dependent on the foraging strategy of the insect.

Emigration. Movement of insects out of the plant population. Immigration back into the plant population was stated in the original model description, but the process was not described. We therefore do not include immigration in our version of the model.

Updating position and assessing quality of the current plant. The reaction of foraging insects to the quality of the different plant types to which they have moved to. Encountering a repellent plant elicits an instant reaction, whereas a deterrent plant causes a delayed reaction in the next time step.

Updating plant status-damage from herbivory. The status of the plants is updated taking into account damage from insect feeding.

Sliding window memory. Insects within the model can remember their previous five plant visitations, this is updated at each time step.

Each insect and plant is processed in order of creation at initialisation of the model. It was unclear from the original model description what order the insects and plants were processed in. After testing with a fixed order and random order, there was no difference in results obtained from the model. Therefore the decision was made to adopt the fixed order. A single time step ends once all agents have been processed as per Fig. 2.

2.1.2. Design concepts

2.1.2.1. Emergence. The emergent behaviour in this model is the spatial distribution and magnitude of damage to different types of plant. The insects make a set of decisions as whether to move, stay or perform a foraging flight dependent on the quality of their current plant (Fig. 2). The ability of the insect to detect its next favourable plant is influenced by the search strategy of the insect (Table 2).

2.1.2.2. Sensing. Insects within the model can access the quality of the plants within their search range. If olfactory or visual searchers this information can be accessed pre-landing (see sub-models section below). If the insects are contact searchers this information can only be accessed post landing and therefore any response is delayed to the next time step. Insects cannot sense the presence of other insects directly, and do not interact. Plants can sense the number of insects present on them, and experience damage through herbivory, adapting the parameter ‘damage threshold’ accordingly.

2.1.2.3. Interaction. No interactions occur between plants, they are simulated as static and non-competing. Interactions between individual insects are indirect. A density dependence is simulated through the depletion of the ‘damage threshold’ of plants which is diminished with each insect visitation. Once the threshold value reaches zero the ‘plant specific flight tendency’ is increased, making it less likely that insects will land and feed upon the plant. Therefore, plants with resources depleted become less attractive to other insects within the model, simulating an indirect density dependent response.

2.1.2.4. Stochasticity. A degree of behavioural stochasticity is present within the model. For each movement, foraging and emigration process there is an associated probability that an insect will make a certain decision (Fig. 2).

2.1.3. Details

2.1.3.1. Initialisation. The parameter values used on initialisation are based on the experiments carried out in the original paper associated with the original model (Potting et al., 2005). Each model run consisted of 50 time steps, with an initial population of 500 insects on a grid of 100 × 100 plants. Insects were given a ‘memory size’ of five plants and ‘insect mortality’ was always set at

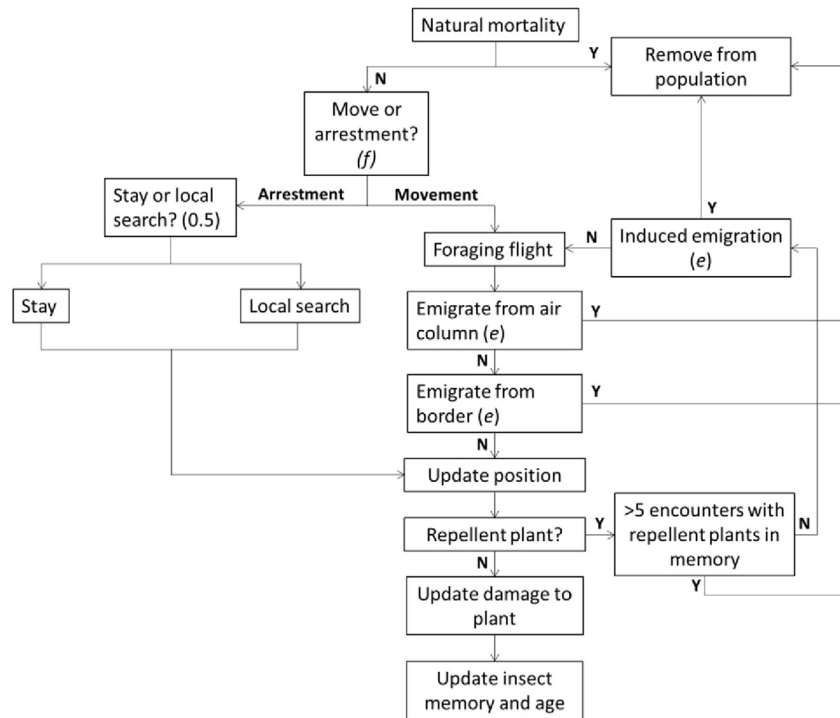


Fig. 2. The sequence of events that happen within one time step of a simulation and for each of the insects in turn. Numbers within brackets describe the probability of that decision happening. Letters within brackets correspond to a parameter in Table 2 that describes the probability of a decision happening. Adapted from Potting et al. (2005).

5%. For possible initialisation parameter values see Table 2.

The initial distribution of the 500 insects can be set as airborne or directional. Airborne insects are distributed randomly in the plant population, with no two insects occupying the exact same point, but potentially the same plant. Insects colonising from a specific direction are given a random x or y coordinate and corresponding x or y coordinate chosen randomly from an exponential distribution with a mean of 0.5.

2.1.3.2. Submodels. For all random decisions within the submodels the probability of a decision being positive or negative was decided by a random number generator selecting a number between zero and one. If the number falls below the probability, the decision was positive, else negative.

Natural mortality. An age dependent mortality rate is applied to each insect at the beginning of each time step. The insects have a probability of surviving equal to:

$$\text{Survival probability} = 1 - (0.001 \times \text{age}) \quad (1)$$

If an insect dies it is removed from the model environment.

Decision to forage. The initial decision to undertake a foraging flight has a probability equal to the 'plant specific flight tendency' of the current plant. If the decision to forage is negative the insect then has a 50% chance it will stay on the current plant and a 50% chance it will hop to a neighbouring plant. The neighbouring plant is chosen at random from the group of eight plants bordering the current location.

Emigration. If the result of the decision to forage is that the insect will perform a foraging flight there is a probability the insect will emigrate whilst in the air column. This probability is set at a default value of 5%. If the insect's present location is at the border of the plant population there is a further 5% chance of emigration from the population. Emigrated individuals are removed from the model environment.

Foraging. Once the insect has decided to initiate a foraging flight the flight proceeds according to the search strategy of the insect. Each insect has a perception distance which forms the area around itself within which it can detect favourable plants. This distance is calculated as a random number between one and the parameter 'maximum movement length' (Table 2). Plants are ranked in their attractiveness to insects, with the lower the 'plant specific flight tendency' the more attractive the plant.

Contact. The insect moves to the plant located at the end of a straight trajectory with length equal to the perception distance. The direction of the trajectory is selected at random, with an angle equal to a random number between zero and 359. Contact searchers can only detect the suitability of a plant post landing, when subject to tactile and gustatory cues.

Visual. The insect can assess plants within a circle of a radius equal to the perception distance. This circle is centred on, but does not include, the current location plant. The insect moves to the closest suitable plant; those with a 'plant specific flight tendency' of 0.05 that is not present in the insect memory of the five previous visitations. If no suitable plants are present, or are present but previously visited, the insect moves to the closest plant of 'plant specific flight tendency' of one (Table 1). This hierarchy of suitable plants was largely assumed as the original model description was ambiguous.

Olfactory. The insect searches the plant cells in a straight line from the current position to the perception distance, starting with the closest. For each suitable plant cell ('plant specific flight tendency' of 0.05 and not present in the memory) there is a 70% chance of landing. It was assumed, as it was not clear in the description, that if the insect did not land on any preferred plants, it landed on the plant at the edge of the search radius.

Assessing quality of plants. Once landed on the chosen plant, if that plant has a 'plant specific flight tendency' of one and is

repellent, the insect has a further set of decisions to make (Table 2). If the number of repellent plants within the memory equals five, the insect emigrates from the plant population and is removed from the model environment. If the repellent plants in the memory totals less than five, the insect has a 5% chance of emigrating from the plant population, then repeats the foraging procedure according to its search strategy. The insect loops through this sequence of events until a non-repellent plant is found. This is an approximation of the original behaviour described in Potting et al. (2005), as information in the original description is ambiguous. It was unclear from the model description what would occur if there are no non-repellent plants in the environment. Within the replica models it was coded that in this case the insects would land on a repellent plant.

Updating age. Insect age increases by one unit each time step.

Updating plant status. Plant damage is calculated cumulatively at each time step. It is recorded by updating the parameter 'cumulative herbivore days' (CHD), which records the number of visitations to a plant per time step. Each time step CHD is updated by adding the total number of insects present on the plant. If CHD causes the 'damage threshold' to fall below zero, the 'plant specific flight tendency' is increased to one, if it does not already equal one.

Updating memory. The insects are simulated with a sliding window memory. The previous five plant visitations are remembered, with the oldest being removed from the memory to make way for the newest visitation.

2.1.3.3. Monitoring. The CHD in crop cells and the location of the insect agents was monitored in each model run. The CHD was totalled and averaged for all crop cells in the environment.

Four simulation scenarios discussed in Potting et al. (2005) were replicated with the NetLogo and Repast Symphony model, henceforth referred to as the NetLogo and Repast models. The models were tested for similarity according to Axtell et al. (1996), who

outline three levels of equivalence: numerical equivalence, models produce results that are numerically identical; distributional equivalence, models produce results that cannot be distinguished statistically and lastly; relational equivalence, parameters within the models have the same relationships, even if there are magnitudinal differences in the dependent variable (Axtell et al., 1996).

A sensitivity analysis of the NetLogo and Repast models was also performed according to methodology presented in Railsback and Grimm (2012). Nine parameters were varied by $\pm 20\%$ (Table 3) and the damage experienced by crop cells monitored. For each simulation scenario the model was run for 50 time steps and repeated 40 times. All statistical analysis was implemented in RStudio Version 0.99.489.

3. Results

3.1. Simulation one: insect sensory mode and displacement speed

The sensory mode of the insect agents was varied to test the impact this may have on the efficacy of the trap crop to prevent damage to crop plants. The planting distribution was set as the trap crop distribution with olfactory, visual and contact searchers. Displacement speed of the insects was varied by increasing or decreasing the 'maximum movement length'. Results are summarised in Fig. 3.

All models report that damage decreases with displacement speed, as insects locate the favourable trap crop more rapidly. The magnitude of damage to crop cells differed between the models, but the rates of reduction in damage with increased movement length did not (multiple pair-wise t-tests with Bonferroni correction, see supplementary material Appendix A, Table A1 for P-values; pairwise comparison of regression coefficients, see supplementary material Appendix A, Table A2 for P-values) (Fig. 3). Despite the differences in magnitude of damage, all three models

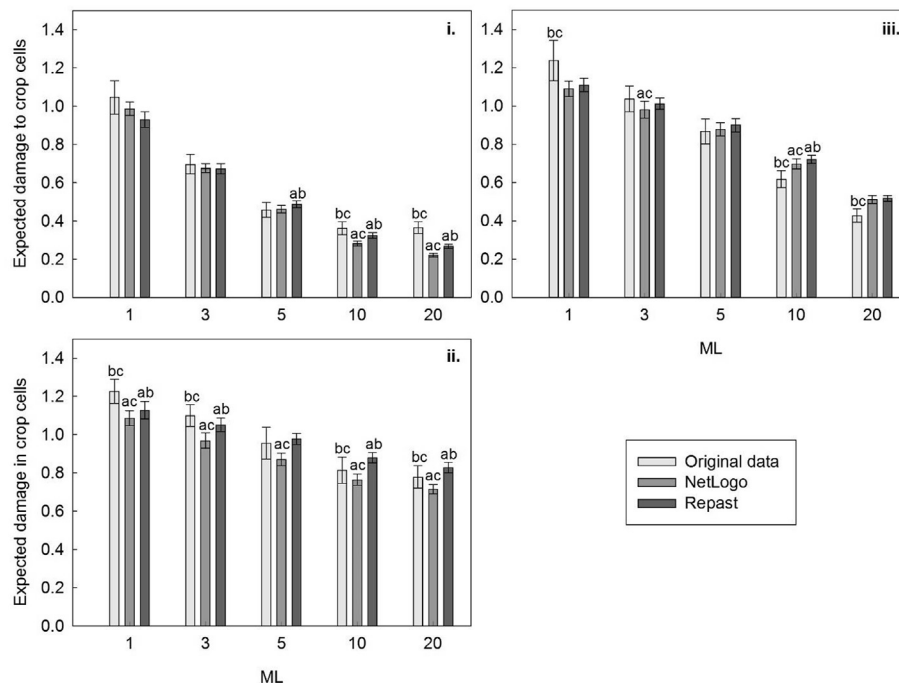


Fig. 3. Total expected damage in crop cells (average of 'cumulative herbivore days' for all 7500 crop cells in the simulation) for each sensory mode (i: Vision; ii: Contact; iii: Olfaction) for a time period of 50 time steps, for the two development platforms (Repast, NetLogo) and original data. 'Maximum movement length' was varied for each strategy (1, 3, 5, 10, and 20). Letters represent statistical significance from alternative development platforms for that given set of parameters (a = Original, b = NetLogo, c = Repast). Plant arrangement was trap crop, N = 40, error bars represent $\pm SD$.

conclude that contact searchers cause the most damage to crop cells, followed by olfactory then visual searchers (Fig. 3).

Transitions between plant cells were recorded during one simulation run of an olfactory searcher in the trap crop environment. ‘Maximum movement length’ was set at 1, 3 or 10. The proportions of these transitions are summarised in Fig. 4.

The proportion of each movement type remained largely similar for both the Repast and NetLogo models, but differed greatly from the original data (Fig. 4). Both models predicted a much higher proportion of crop to crop movements and a much lower proportion of trap to trap movements (Fig. 4).

3.2. Simulation two: colonisation pattern

This simulation tested the effects of varying colonisation pattern on the efficacy of different spatial arrangements of trap crop plants to minimise damage to crop plants. The colonisation pattern of insects was set as either random, to replicate airborne insects, or directional along one edge of the simulated plant population. The spatial distribution of plants was set as either border, trap crop or patches (Fig. 1). Results are summarised in Fig. 5.

For the border planting arrangement the Repast and NetLogo models predicted near zero levels of damage to crop cells when insects colonise from the northern or eastern edges, in contrast to the original data (Multiple pair-wise t-tests, see [supplementary materials](#) Appendix 2 Table A3 for P-values, Fig. 5i). The NetLogo model predicted the least damage for these scenarios (Multiple pair-wise tests, see [Appendix A](#), Table A3 for P-values, Fig. 5i).

For the random patch distribution the reproduced models produced near identical data that differed in magnitude to the original, with NetLogo and Repast models predicting lower levels of damage when insects colonised from a specific direction, compared to the predictions of the original model (Multiple pair-wise t-tests, see [Appendix A](#), Table A3 for P-values, Fig. 5ii).

For the intercrop distribution the replicated models predicted

levels of damage that differed with the original data, for each colonisation direction (Multiple pair-wise t-tests, see [Appendix 2](#) Table A3 for P-values, Fig. 5). Notably the NetLogo and Repast predicted much higher levels of crop damage when insects colonised from the East, compared to the original model (Multiple pair-wise t-tests, see [Appendix A](#), Table A3 for P-values, Fig. 5iii).

3.3. Simulation three: indirect density dependent effects

The effects of indirect density dependence in the trap crop were tested by varying the ‘damage threshold’. The ‘damage threshold’ was set at values far lower than the crop cells (1, 3 and 5) and then equal to the crop cells (100). A lower ‘damage threshold’ results in a larger number of plants becoming repellent to insects when their resources are depleted. This increases the likelihood that insects will leave the trap crop plants and re-enter the crop.

When plants were arranged in an intercrop environment and ‘damage threshold’ varied, the original and reproduced model data differed substantially in magnitude (Fig. 6i). All but one parameter combination resulted in damage predictions that differed significantly from model to model (Multiple pairwise t-tests, see [supplementary material](#) Appendix A, Table A4 for P-values; Fig. 6i).

The NetLogo and Repast models differed in the trend of the decrease in crop damage with increasing ‘damage threshold’ (Pairwise t-test comparison of regression coefficients: $t = -5.63$, $p < 0.05$, $df = 4$, see [supplementary material](#) Appendix A Table A5 for additional P-values), although the Repast and original data did not.

For the border planting arrangement, again the NetLogo model produced results that differed in rate of decrease of damage with ‘damage threshold’ compared to the other models (Pairwise t-test comparison of regression coefficients: Repast- $t = -5.23$, $p < 0.05$, $df = 4$, Original- $t = -5.85$, $p < 0.05$, $df = 4$, see [supplementary material](#) Appendix A, Table A5 for additional P-values). The NetLogo model predicted very little reduction in crop cell damage

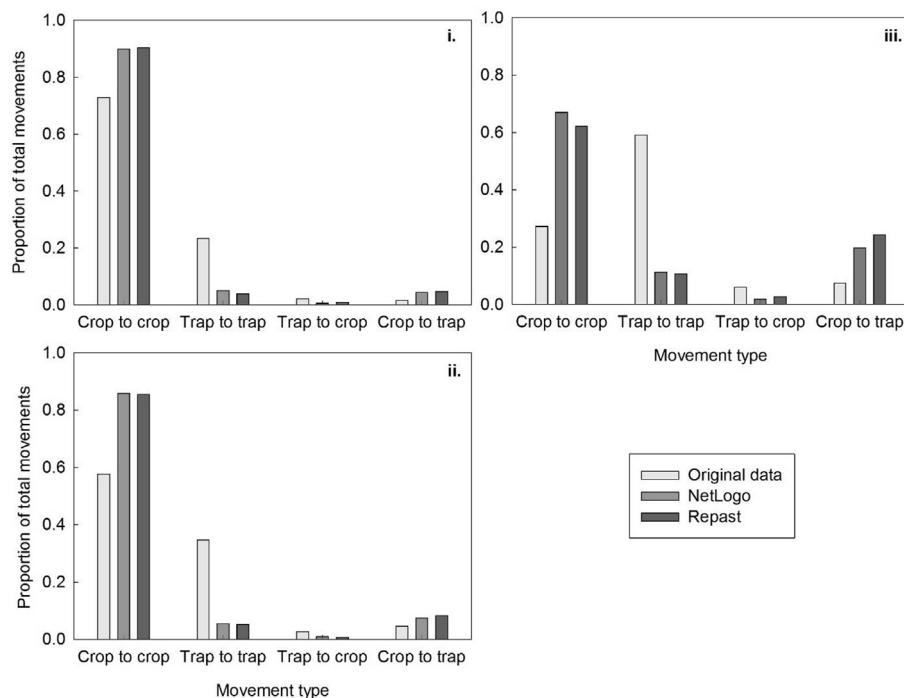


Fig. 4. The proportion of transition events between crop and trap crop cells for one run of the simulation, for each development platform. ‘Maximum movement length’ was varied from 1 (i.), 3 (ii.) and 10 (iii.) for an olfactory searcher in a trap crop environment.

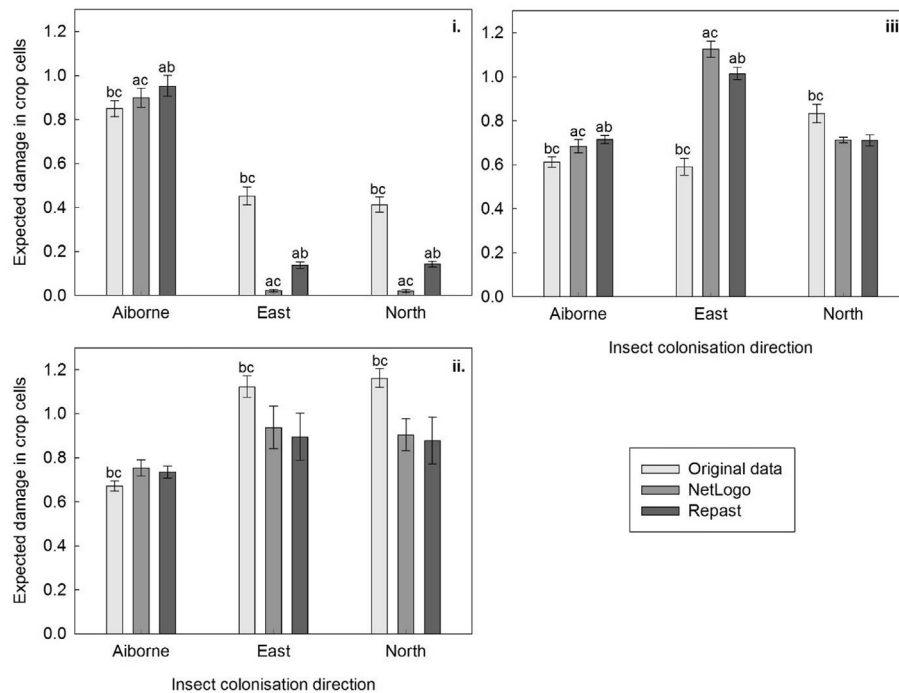


Fig. 5. Total expected damage in crop cells for three spatial distributions of trap and crop plants (i: Border, ii: Patches and iii: Intercrop.) Total expected damage calculated as average of 'cumulative herbivore days' for all crop cells in the simulations (Intercrop and Patches: 7500, Border: 7396) when insects colonise from the east, north or randomly through airborne dispersal. Simulated insects were olfactory searchers with a 'maximum movement length' of 10. Simulations ran for a time period of 50 time steps, for the two development platforms (Repast, NetLogo) and original data. Letters represent statistical significance from alternative development platforms for that given set of parameters (a = Original, b = NetLogo, c = Repast). N = 40, error bars represent \pm SD.

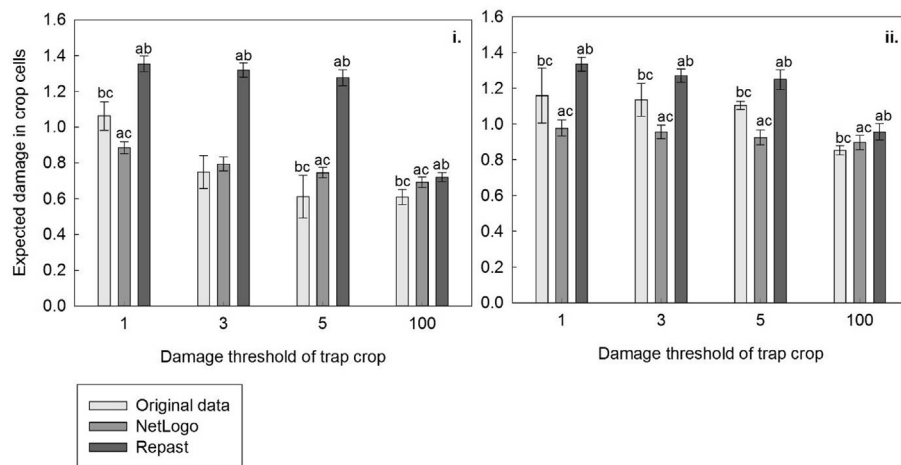


Fig. 6. The effects of varying density dependent effects in the trap crop cells, modelled as varying 'damage threshold' for trap crop (1, 3, 5, 100) in an intercrop (i.) and border (ii.) environment. Expected damage to crop cells was calculated as mean number of 'cumulative herbivore days' for each crop cell in the simulation (Intercrop = 7500 Border = 7396). Simulated insects were olfactory searchers with a maximum movement length of 10, colonising at random. Letters represent statistical significance from alternative development platforms for that given set of parameters (a = Original, b = NetLogo, c = Repast). N = 40, error bars represent \pm SD.

when trap 'damage threshold' was increased (Fig. 6ii). Each model predicted damage at significantly different magnitudes, for all parameter combinations (Multiple pairwise t-tests with Bonferroni correction, see [supplementary material](#) Appendix A, Table A4 for P-values, Fig. 6).

3.4. Simulation four: deterrent/repellent trap crop plants

The effect of including within the planting design a non-crop plant that actively repelled or deterred insects was tested by

changing the qualities of the trap crop cells in the intercrop planting pattern. Repellent plants actively repel the insects within a time step until they land on a non-repellent plant, also increasing their likelihood of emigrating with each flight. Deterrent plants increased the probability that the insect would emigrate in the next time step; a delayed response. This induced emigration rate was varied.

The generated data from the three models differed in magnitude (Multiple pairwise t-tests, see [supplementary material](#) Appendix A Table A6 for P-values, Fig. 7). When trap plants were set as

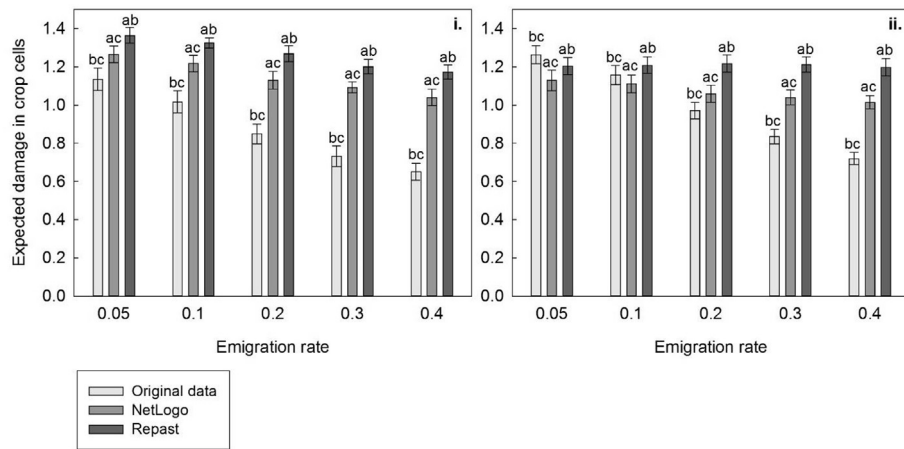


Fig. 7. Total expected damage in crop cells (average of ‘cumulative herbivore days’ for all 7500 crop cells in the simulation) when trap crop cells within the intercrop planting design were set as deterrent (i.) or repellent (ii.) for a time period of 50 time steps, for the two development platforms (Repast, NetLogo) and original data. The probability of emigration caused by encounters with repellent or deterrent plants was varied (0.05, 0.1, 0.2, 0.3 and 0.4). Simulated insects were olfactory searchers with a ‘maximum movement length’ of 10, colonising at random. Letters represent statistical significance from alternative development platforms for that given set of parameters (a = Original, b = NetLogo, c = Repast). $N = 40$, error bars represent $\pm SD$.

deterrent, both NetLogo and Repast models produced data that decreased in damage at very similar rates, but differed from the steeper decline of the original model (Pairwise t -test comparison of regression coefficients: Repast- $t = -5.45$, $p < 0.05$, $df = 6$, NetLogo $t = -4.73$, $p < 0.05$, $df = 6$, see [supplementary material](#) Appendix A Table A7 for additional P-values, [Fig. 7i](#)).

It is a similar case when trap plants were set to repellent, all models produced damage predictions that differed significantly in magnitude (Multiple pairwise t -tests, see [supplementary material](#) Appendix A Table A6 for P-values, [Fig. 7ii](#)). All three models also differed in the pattern of the data, the original model predicting the steepest decline in damage with increasing emigration probability and the Repast model the least (Pairwise t -test comparison of regression coefficients, see [supplementary material](#) Appendix A Table A7 for P-values, [Fig. 7ii](#)).

3.5. Local sensitivity analysis of NetLogo and repast model

A sensitivity analysis of eight parameters was performed on the reproduced NetLogo and Repast models, for each of the insect search strategies in an intercrop environment (unless otherwise stated) ([Table 3](#)). Each parameter was varied $\pm 20\%$ and the S value of the magnitude of change in the response variable was calculated as the difference between the S+ and S- score, according to [Railsback and Grimm \(2012\)](#). See [Table 3](#), for a summary of the

analysis.

Despite being written with the same set of rules, the two models responded differently to the variation of the parameters. This is most evident for the contact searchers ([Table 3](#)), with the biggest disparity in response being seen for the parameters: ‘damage threshold’ in main crop and insect mortality rate. For olfactory searchers the sensitivity scores were marginally more similar, but the NetLogo model was still much more sensitive to changes in the ‘damage threshold’ of both plant types and emigration probability than the Repast model ([Table 3](#)). The visual searchers produced the closest similarity scores for the two models, but there are still differences in the level of sensitivity ([Table 3](#)).

4. Discussion

The replication process resulted in three models that are numerically, distributionally and relationally dissimilar ([Axtell et al., 1996](#)). This appears to be a result of reinterpreting the complex model in different platforms using different programming languages and styles, complicated by lacking information in the original model description. The initial replication was written using the NetLogo platform following the model description alone, which described a model written in Visual Basic ([Potting et al., 2005](#)). The original code was not available from the publication, or from contact with the authors (personal communication, 2014). The

Table 3

Summary of local sensitivity analysis on the NetLogo and Repast model, S scores are reported for each parameter (varied by $\pm 20\%$) and for each sensory mode for both models.

Parameter	Contact		Olfactory		Visual	
	NetLogo	Repast	NetLogo	Repast	NetLogo	Repast
	S	S	S	S	S	S
Plant specific flight tendency (trap crop)	0.091	0.054	0.011	0.049	0.005	0.005
Damage threshold main crop	0.019	0.131	0.094	0.034	0.000	0.011
Damage threshold trap crop	0.033	0.070	0.053	0.001	0.006	0.001
Emigration tendency	0.066	0.141	0.074	0.008	0.020	0.002
Maximum movement length	0.171	0.135	0.102	0.070	0.093	0.077
Insect mortality	0.020	0.116	0.059	0.044	0.038	0.022
Percentage of trap crop in environment (simulated as changing the number of randomly placed 10×10 patches)	0.053	0.008	0.050	0.052	0.094	0.123
Probability of landing in flight (olfactory searchers only)	N/A	N/A	0.079	0.053	N/A	N/A

differences between the original and NetLogo model are most likely due to misunderstandings stemming from interpretation of the model description.

There were two obvious misinterpretations in the replicated model. The first when considering Simulation 2 (Fig. 5iii). The damage to crop cells in the intercrop scenario differs to the original data for the eastern edge colonisation, with the original data predicting much higher levels of damage to crop plants. This is most likely a result of a difference in interpretation of the spatial distribution of the plants. Following the description given in Potting et al. (2005), insects colonising from the east of the intercrop environment would encounter a band of non-host crop plant cells first, before encountering the first band of host trap-crop cells (Fig. 1). This results in the higher levels of damage produced by the NetLogo and Repast models, but does not explain the lower levels of damage predicted by the original data (Fig. 5iii). When the same simulation is implemented with colonisation from the west, the damage to crop cells is found to be lower due to the insects encountering a strip of trap crop cells first, producing results that are closer to the original data (data not shown).

The second misinterpretation is a general lack of understanding of the sub-models controlling insect movement within the original model. This is most clear when considering Simulation 4. The interpretation of the repellent/deterrent rule is quite obviously different from the original model, with the crop damage levels in the original model much more dependent on the rate of induced emigration from unsuitable plants (Fig. 7). From the model description and flow diagram provided in the original publication it is unclear which aspect of this rule has been confused (Potting et al., 2005).

Lacking information in model descriptions is a known problem with agent based models (Müller et al., 2014; Polhill et al., 2008), and a recognised issue when implementing model replication (Axtell et al., 1996). There are a number of proposed frameworks for addressing this, with the aim of ensuring that adequate information is supplied to aid model replication and understanding (Müller et al., 2014). Perhaps the most popular of these frameworks is the ODD (Overview, Design concepts and Details) protocol proposed by Grimm et al. (2006) (Müller et al., 2014; Polhill et al., 2008; Thiele and Grimm, 2015), which has been found to greatly improve the level of information that is communicated alongside models (Polhill et al., 2008).

The original description was not written following any of the frameworks now proposed for agent based models. Had the authors adopted the ODD protocol (it should be noted that the protocol was developed post publication), interpretation of the model may have been less problematic. However, there were complex misunderstandings that occurred during the replication process that were less likely to be remediated by improving the model description. An adequate description of the submodels controlling the insect movement would be extremely lengthy and convoluted, and could be much more succinctly communicated by providing code. This leads us to the conclusion that working code would be needed to fully interpret a complex agent based model such as this. The description can provide important clarification of the aims, purposes and theory of the models, but should never be relied upon as the sole source of information to enable replication.

However, when reproducing the NetLogo model in Repast using the working NetLogo code, access to working source code was no guarantee of a closely equivalent replication. Despite adhering to the same set of rules and parameters the model results differed in magnitude of predicted damage and also in the trends of the data. Again, the sub-models controlling insect movement were sources of variation between the data generated by the two models. The coding of the insect avoidance behaviour of non-acceptable plants

appeared to be a large source of variation. The NetLogo model simulated a decrease in crop damage with an increase in the 'damage threshold' of the trap crop, whereas the Repast model was not sensitive to this parameter despite both models having near identical scheduling patterns (Fig. 6). Additionally, the local sensitivity analysis was particularly telling, with both models having very different sensitivity scores for a number of parameters (Table 3). The differences between the models were not trivial and relationships between the parameters differed substantially.

Differences in the data produced by the NetLogo and Repast models appear to be a result of subtle differences in the programming languages used and the style of coding that had to be adopted (Railsback et al., 2006). NetLogo is a very different programming language to Java (Railsback et al., 2006). NetLogo involves regular use of high level structures and primitives, whereas using Java in the Repast environment requires a more frequent use of low level functions to simulate the insect/plant behaviour. The Java code was much more verbose than the NetLogo code, with more aspects of the plant and insect behaviour having to be manually defined. This resulted in subtle differences in how both models were scheduled and movement was encoded. It is possible that this is the underlying cause of many of the differences between the two models. Very similar issues were experienced by Bajracharya and Duboz (2013) when replicating a simple agent based epidemiological model, although to much less an extent.

The model we chose to replicate is a complex representation of a multispecies interaction in an agro-ecological setting. When replicating complex models across languages and platforms it is likely that inspecting all of the interacting consequences of the coding choices made due to the constraints or style of a language or platform would be an arduous task, unlikely to be carried out by most modellers. Secondly, had the second replication in Java not been implemented, there would have been no indication that the variation in model results could have been a result of translational differences. This highlights the need to proceed with caution when replicating from description alone, and the need to be aware of translation as a possible source of variation.

The replication in this case was not completely successful and the implications of this were that we were unable to test the robustness of this model as it was presented in the original publication. We were unable to reuse the model to test the wider parameter space of the existing simulations, which is one major benefit of replicating computational models. The magnitudinal differences in the damage predicted by the three models was realistically of limited consequence. This model is broadly predictive and the scale of damage level used is arbitrary. The differences in the trends of the results when parameters were varied rendered the model irreproducible, as very different conclusions were drawn from the replicated model compared to the original. Although we were unable to test the original experiments, we were still able to develop usable models to test our own research questions based on the concepts outlined by Potting et al. (2005), and present the models for further use and development.

The problems with replication experienced here may be of greater consequence for models generating more precise predictions and is most certainly not just limited to ABMs, or ABMs of this nature. For example, process based models, highly precise climate models and models with a high proportion of stochasticity, among others have reported confusion in results when replicated (Bocedi and Travis, 2016; Easterbrook, 2014; Seppelt and Richter, 2005). The precise results produced by hydrological and complex land use ABMs would also suggest they may be susceptible to produce differing results when replicated across languages or platforms (Bithell and Brasington, 2009; Moglia et al., 2010). Implementing multiple replications across platforms and

languages, then comparing model results, would be essential to assess the repeatability of any model that utilises a specific programming language and/or development platform regardless of the modelling approach or subject matter.

5. Conclusions

Our findings add to an emerging body of literature stating the very valid argument that access to source code is paramount (Axtell et al., 1996; Grimm, 1999; Grimm et al., 2005; Thiele and Grimm, 2015). Encouraging use of repositories such as the openABM library, which enables easy access to source code and model descriptions could ensure future ABMs are more replicable and reusable (Janssen et al., 2008). However, cross platform/language replication must be considered as a potential source of variability in model results. There is the potential for cross platform and language differences that may fundamentally change the outcomes and conclusions drawn from the replicated model compared to the original. Increased model reuse, testing and replication is needed to ensure agent based modelling is appreciated as a valuable research method. Additionally, the implications of this study apply to any modelling methodology that utilises a specific programming language. Building a body of replication literature, addressing cross platform/language replication will ensure that researchers are aware of the potential pitfalls and will strengthen our understanding of all methods of modelling in the sciences (Hales et al., 2003).

Acknowledgements

This project was part funded by an Institute of Biological, Environmental and Rural Sciences (IBERS), Aberystwyth University, PhD studentship. We are grateful for access to BBSRC-funded facilities in IBERS during the project. The manuscript was improved through discussion with J. Kiukas.

Appendix A. Supplementary data

Supplementary data related to this article can be found at <http://dx.doi.org/10.1016/j.envsoft.2017.01.020>.

References

- Almeida, S.J., de Martins Ferreira, R.P., Eiras, Á.E., Obermayr, R.P., Geier, M., 2010. Multi-agent modeling and simulation of an *Aedes aegypti* mosquito population. *Environ. Model. Softw.* 25, 1490–1507. <http://dx.doi.org/10.1016/j.envsoft.2010.04.021>.
- Axelrod, R., 1997. Advancing the art of simulation in the social sciences. In: Conte, R., Hegselmann, R., Terna, P. (Eds.), *Simulating Social Phenomena*. Springer-Verlag, pp. 21–40. http://dx.doi.org/10.1007/978-3-662-03366-1_2.
- Axtell, R., Axelrod, R.M., Epstein, J.M., Cohen, M.D., 1996. Aligning simulation models: a case study and results. *Comput. Math. Organ. Theory* 1, 123–141. <http://dx.doi.org/10.1007/BF01299065>.
- Bajracharya, K., Duboz, R., 2013. Comparison of three agent-based platforms on the basis of a simple epidemiological model (WIP). In: *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative M&S Symposium*, p. 6.
- Bithell, M., Brasington, J., 2009. Coupling agent-based models of subsistence farming with individual-based forest models and dynamic models of water distribution. *Environ. Model. Softw.* 24, 173–190. <http://dx.doi.org/10.1016/j.envsoft.2008.06.016>.
- Bocedi, G., Travis, J.M.J., 2016. Models of dispersal evolution highlight several important issues in evolutionary and ecological modeling. *Am. Nat.* 187 <http://dx.doi.org/10.1086/684191>, 000–000.
- Bradhurst, R.A., Roche, S.E., East, I.J., Kwan, P., Garner, M.G., 2016. Improving the computational efficiency of an agent-based spatiotemporal model of livestock disease spread and control. *Environ. Model. Softw.* 77, 1–12. <http://dx.doi.org/10.1016/j.envsoft.2015.11.015>.
- Collins, A., Petty, M., Vernon-Bido, D., Sherfey, S., 2015. A call to arms: standards for agent-based modeling and simulation. *J. Artif. Soc. Soc. Simul.* 18, 1–15. <http://dx.doi.org/10.18564/jasss.2838>.
- Easterbrook, S.M., 2014. Open code for open science? *Nat. Geosci.* 7, 779–781. <http://dx.doi.org/10.1038/ngeo2283>.
- Evans, M.R., Grimm, V., Johst, K., Knuuttila, T., de Langhe, R., Lessells, C.M., Merz, M., O'Malley, M.A., Orzack, S.H., Weisberg, M., Wilkinson, D.J., Wolkenhauer, O., Benton, T.G., 2013. Do simple models lead to generality in ecology? *Trends Ecol. Evol.* 28, 578–583. <http://dx.doi.org/10.1016/j.tree.2013.05.022>.
- Grimm, V., 1999. Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecol. Modell.* 115, 129–148. [http://dx.doi.org/10.1016/S0304-3800\(98\)00188-4](http://dx.doi.org/10.1016/S0304-3800(98)00188-4).
- Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S.K., Huse, G., Huth, A., Jepsen, J.U., Jørgensen, C., Mooij, W.M., Müller, B., Pe'er, G., Piu, C., Railsback, S.F., Robbins, A.M., Robbins, M.M., Rossmanith, E., Rüger, N., Strand, E., Souissi, S., Stillman, R.A., Vabø, R., Visser, U., DeAngelis, D.L., 2006. A standard protocol for describing individual-based and agent-based models. *Ecol. Modell.* 198, 115–126. <http://dx.doi.org/10.1016/j.ecolmodel.2006.04.023>.
- Grimm, V., Berger, U., DeAngelis, D.L., Polhill, J.G., Giske, J., Railsback, S.F., 2010. The ODD protocol: a review and first update. *Ecol. Modell.* 221, 2760–2768. <http://dx.doi.org/10.1016/j.ecolmodel.2010.08.019>.
- Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W.M., Railsback, S.F., Thulke, H.-H., Weiner, J., Wiegand, T., DeAngelis, D.L., 2005. Pattern-oriented modeling of agent-based complex systems: lessons from ecology. *Science* 310, 987–991. <http://dx.doi.org/10.1126/science.1116681>.
- Hales, D., Rouchier, J., Edmonds, B., 2003. Model - to - model analysis. *J. Artif. Soc. Soc. Simul.* 6, 1–5.
- Janssen, M.A., Alessa, L.N., Barton, M., Bergin, S., Lee, A., 2008. Towards a community framework for agent - based modelling. *J. Artif. Soc. Soc. Simul.* 11, 1–12.
- Matthews, R.B., Gilbert, N.G., Roach, A., Polhill, J.G., Gotts, N.M., 2007. Agent-based land-use models: a review of applications. *Landsc. Ecol.* 22, 1447–1459. <http://dx.doi.org/10.1007/s10980-007-9135-1>.
- Meyerovich, L. a., Rabkin, A.S., 2013. Empirical analysis of programming language adoption. In: *Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications - OOPSLA '13*, pp. 1–18. <http://dx.doi.org/10.1145/2509136.2509515>.
- Moglia, M., Perez, P., Burn, S., Moglia, M., Perez, P., Burn, S., 2010. Modelling an urban water system on the edge of chaos. *Environ. Model. Softw.* 25, 1528–1538. <http://dx.doi.org/10.1016/j.envsoft.2010.05.002>.
- Müller, B., Balbi, S., Buchmann, C.M., de Sousa, L., Dressler, G., Groeneveld, J., Klassert, C.J., Le, Q.B., Millington, J.D.A., Nolzen, H., Parker, D.C., Polhill, J.G., Schlüter, M., Schulze, J., Schwarz, N., Sun, Z., Taillandier, P., Weise, H., 2014. Standardised and transparent model descriptions for agent-based models: current status and prospects. *Environ. Model. Softw.* 55, 156–163. <http://dx.doi.org/10.1016/j.envsoft.2014.01.029>.
- North, M.J., Collier, N.T., Ozik, J., Tataru, E.R., Macal, C.M., Bragen, M., Sydelko, P., 2013. Complex adaptive systems modeling with repast symphony. *Complex Adapt. Syst. Model.* 1, 1–26. <http://dx.doi.org/10.1186/2194-3206-1-3>.
- Perez, L., Dragicevic, S., 2010. Modeling mountain pine beetle infestation with an agent-based approach at two spatial scales. *Environ. Model. Softw.* 25, 223–236. <http://dx.doi.org/10.1016/j.envsoft.2009.08.004>.
- Polhill, J.G., Parker, D.C., Brown, D.G., Grimm, V., 2008. Using the ODD protocol for comparing three agent-based social simulation models of land use change. *J. Artif. Soc. Soc. Simul.* 11, 1–31.
- Potting, R.P.J., Perry, J.N., Powell, W., 2005. Insect behavioural ecology and other factors affecting the control efficacy of agro-ecosystem diversification strategies. *Ecol. Modell.* 182, 199–216. <http://dx.doi.org/10.1016/j.ecolmodel.2004.07.017>.
- Railsback, S., Grimm, V., 2012. *Agent-based and Individual-based Modeling: a Practical Introduction*. Princeton University Press, Princeton.
- Railsback, S.F., Lytinen, S.L., Jackson, S.K., 2006. Agent-based simulation platforms: review and development recommendations. *Simulation* 82, 609–623. <http://dx.doi.org/10.1177/0037549706073695>.
- Rollins, N.D., Barton, C.M., Bergin, S., Janssen, M.A., Lee, A., 2014. A computational model library for publishing model documentation and code. *Environ. Model. Softw.* 61, 59–64. <http://dx.doi.org/10.1016/j.envsoft.2014.06.022>.
- Seppelt, R., Richter, O., 2005. "It was an artefact not the result": a note on systems dynamic model development tools. *Environ. Model. Softw.* 20, 1543–1548. <http://dx.doi.org/10.1016/j.envsoft.2004.12.004>.
- Sun, Z., Lorscheid, I., Millington, J.D., Lauf, S., Magliocca, N.R., Groeneveld, J., Balbi, S., Nolzen, H., Müller, B., Schulze, J., Buchmann, C.M., 2016. Simple or complicated agent-based models? A complicated issue. *Environ. Model. Softw.* 86, 56–67. <http://dx.doi.org/10.1016/j.envsoft.2016.09.006>.
- Thiele, J.C., Grimm, V., 2015. Replicating and breaking models: good for you and good for ecology. *Oikos* 124, 691–696. <http://dx.doi.org/10.1111/oik.02170>.
- Tisue, S., Wilensky, U., 2004. NetLogo: a simple environment for modeling complexity. In: *International Conference on Complex Systems*, pp. 1–10. <http://dx.doi.org/10.1109/ICVD.2004.1261037>.
- Wilensky, U., Rand, W., 2007. Making models match: replicating an agent-based model. *J. Artif. Soc. Soc. Simul.* 10, 1–24.